

PREDICTING POLL TRENDS WITH FACIAL EMOTIONS:  
A NEURAL NETWORK APPLIED TO EMOTIONS FROM  
PHOTOS OF POLITICIANS IN THE ONLINE PRESS

MASTER THESIS PAPER

Victor Cluzel 525700

Reviewers: Prof. Dr. Stefan Lessmann  
Prof. Dr. Wolfgang Karl Härdle

filed on January 25, 2018

# Table of Contents

<b>List of Figures</b>	<b>IV</b>
<b>List of Tables</b>	<b>IV</b>
<b>1 Introduction</b>	<b>2</b>
<b>2 Literature Review</b>	<b>3</b>
2.1 Emotion in politics . . . . .	3
2.2 Sentiment analysis & political outcomes . . . . .	4
2.3 Emotion extraction . . . . .	5
<b>3 Motivation</b>	<b>6</b>
<b>4 Descriptive Statistics</b>	<b>8</b>
4.1 Emotions : a pattern for each candidate . . . . .	9
4.2 Difference between upward trends and downward trends . . . . .	10
<b>5 Data Preparation</b>	<b>11</b>
<b>6 Model Creation &amp; Optimization</b>	<b>13</b>
6.1 Model Creation . . . . .	14
6.2 Model Selection and Parametrization . . . . .	14
<b>7 Empirical Results</b>	<b>16</b>
7.1 Variable importance . . . . .	17
7.2 Presentation of final results . . . . .	18
<b>8 Discussion</b>	<b>19</b>
<b>9 Conclusion</b>	<b>20</b>
<b>Appendix</b>	<b>25</b>
<b>A Image Database creation</b>	<b>25</b>
A.1 The Google Custom Search API . . . . .	25
A.2 Getting the initial image database . . . . .	25

<b>B</b>	<b>Data Cleaning &amp; Emotion Recognition</b>	<b>26</b>
B.1	Necessity of a "manual" cleaning . . . . .	26
B.2	The Microsoft Emotion API . . . . .	26
B.3	Final image database & emotion database creation . . . . .	27
<b>C</b>	<b>More on Descriptive Statistics</b>	<b>29</b>
<b>D</b>	<b>Table of Variables</b>	<b>31</b>
<b>E</b>	<b>Neural Network Programming Details</b>	<b>32</b>
E.1	The Keras Module . . . . .	32
E.2	The Hyperopt Module . . . . .	32
<b>F</b>	<b>Multi Layer Perceptron Model</b>	<b>33</b>
<b>G</b>	<b>Training the MLP : additional plots</b>	<b>34</b>

## List of Figures

1	IPSOS SOPRA STERIA poll results . . . . .	6
2	Emotion distribution for all the candidates . . . . .	8
3	Comparison of the emotion distribution for each candidate . . . . .	9
4	Comparison of emotion distributions between upward trends and downward trends . . . . .	10
5	Comparison of main emotion distributions between upward trends and downward trends . . . . .	11
6	PCA of data before standardization . . . . .	12
7	PCA of data after standardization . . . . .	13
8	Confusion matrix of the MLP on 17 variables . . . . .	17
9	ROC curve of the MLP on 15 variables . . . . .	19
10	Example of output from the Microsoft Emotion API . . . . .	26
11	Comparison of the main emotion distribution for each candidate . . . . .	29
12	Emotion distribution for Emmanuel Macron . . . . .	29
13	Emotion distribution for Jean-Luc Mélenchon . . . . .	29
14	Emotion distribution for Marine Le Pen . . . . .	30
15	Emotion distribution for Benoît Hamon . . . . .	30
16	Emotion distribution for François Fillon . . . . .	30
17	Multi Layer Perceptron Model scheme (trained on 17 variables) . . . . .	33
18	Evolution of the accuracy during the training of the 15 variables MLP . . . . .	34
19	Evolution of the sparse categorical crossentropy during the training of the 15 variables MLP . . . . .	34

## List of Tables

1	Resulting parameters from the optimizations . . . . .	15
2	Results of the initial MLP on the 17 variables. . . . .	16
3	Results of the first step of the wrapper . . . . .	18
4	Results of the final MLP on the 15 selected variables. . . . .	18
5	Table of images downloaded and remaining after manual cleaning . . . . .	28
6	Table of initial variables and treatment applied . . . . .	31

## **Abstract**

Little attention has been paid so far on applying facial emotion recognition to politics. Is it possible to highlight a link between emotion detected on candidates pictures and voting intentions in a political election ? Trying to answer this question, we use the recent French 2017 presidential election as a case study to predict poll trends based on the emotions in pictures of the candidates in the press. The Microsoft Emotion API has been put to use in order to provide the emotional data as a basis for our study. A Multi Layer Perceptron, a type of Artificial Neural Network is implemented to predict the poll slope sign. We find that there is a moderate correlation in our sample between the concepts of facial emotions of political candidates and poll trends, our algorithm reaching 64% accuracy.

## **Acknowledgements**

I would like to thank all the following people, that made possible to achieve the research and the writing of this Master Thesis:

Prof. Dr. Stefan Lessmann, who helped identify the precise application of my research and followed my Master Thesis,

Prof. Dr. Wolfgang K. Härdle, who gave me the idea to use the Microsoft Emotion API and provided useful insight on my research,

Maxence Ernoult, PhD Candidate, who has been able to give me some insights on the theory behind Artificial Neural Networks,

Bruno Spilak, who has been very helpful in understanding the way modules Keras and Hyperopt have to be installed and used,

My manager and colleagues at the French Directorate-General of the Treasury, who adapted my schedule to allow me to finish my Master Thesis,

And all of my family and friends, who patiently waited for me to finish my Master Thesis to be able to meet me again.

# 1 Introduction

"The greatest value of a picture is when it forces us to notice what we never expected to see."

This statement by famous twentieth century statistician John Tukey puts into perspective the effect of visual representation of people in the press. Indeed, when studying the impact of a piece of news on the reader, we have to take into account that above the general sentiment drawn from the article when it is read, the first things that notice he who open a newspaper -whether it is online or real- is the picture. So how to interpret this effect ?

Our study will focus on politicians and more specifically their emotions in those pictures. This choice has been made because it is possible to find a proxy for effect on people of these emotions : poll results. They are of course not a direct consequence of the perception of pictures in the press, but they would likely have a link -and not necessarily a causal effect- with these emotions. In fact, one can suppose that when seeing such emotion on the face of an election candidate, whether it is sadness, happiness or neutrality, voters may be more likely to vote for him. Reversing the thinking, poll result may also have an effect on the candidate emotions. Even though they are given tremendous advice by their communications manager to show the emotions they decide to show, we can assume that press photographs also capture and publish the emotions they didn't decide to show and which could be assumed more real<sup>1</sup>.

In order to estimate this link between emotions and poll results, the following model has been implemented. Just after the 2017 presidential elections in France, the Google Custom Search<sup>2</sup> API<sup>3</sup> has allowed to download a set of images related to this election. Subsequently, after the image data has been cleaned, the Microsoft Emotion<sup>4</sup> API provides the emotional data for the candidates, predicting it from

---

<sup>1</sup>The eventual journalistic bias is here not analyzed, that's why the link is studied between polls and emotions represented on photos of politicians in the news, and not directly the emotions of politicians. Indeed what most of the public opinion sees are the photos and videos of the candidates and not directly their faces.

<sup>2</sup>Licensed under Creative Commons CC BY.

<sup>3</sup>Applications Programming Interface.

<sup>4</sup>Licensed under the MIT License.

the previous images. Once again the data is cleaned in order to apply on it a Multi Layer Perceptron (MLP), a specific Neural Network. The output of the model is a vector that predicts whether the trend in the polls for each candidate goes upwards or downwards.

After an overview of the literature on emotion extraction, the use of analytics in politics and more specifically Neural Networks, we will shortly present the choices that have been made while building and cleaning the image database. Then the descriptive Statistics of the final numerical database of emotions will show a few stylized facts on the correlation between emotions of politicians and poll results. At last, MLP results will be presented and discussed with comparison with another predictive algorithm, along with the eventual sociological and political outcome of this paper implications.

## 2 Literature Review

### 2.1 Emotion in politics

In social sciences, it used to be difficult to find good predictors for behavior, as said Thomas [Scheff \(2005\)](#) in his speech on Emotions and Politics. In his numerous works ([Scheff, 2003](#)), this Professor nevertheless frequently argues that for instance shame is a good predictor, and a very powerful feeling motivating a lot of public actions. But it is an invisible emotion, as it is a recursive one : the shame of shame tends to make the shame hide itself. So how is it possible to use other emotions, those that we can see or guess on others, to predict social issues and especially political ones ?

Alison [Powell \(2016\)](#) warns about the assumptions underpinning sentiment analysis. Nowadays algorithms are presenting the world to us: *via* the algorithms-ruled social networks or the referencing of search engines for instance. Those algorithms are often computing sentiment analysis, consequently focusing on the way people emote about the facts that are presented to them. In such a scheme, one can wonder how emotion contagion can spread quickly when dealing with politics. And this also raises the question of the rise of "fact-free" politics: is it linked to these algorithms ? In this perspective, it would be interesting to be able to pinpoint the link between

political emotions and public opinion through image recognition.

## 2.2 Sentiment analysis & political outcomes

Recently, artificial intelligence became more and more developed, and thus enabling us to create predictors using larger and larger databases.

[Searles and Ridout \(2017\)](#) examined the use of emotional appeal during the 2016 U.S. presidential campaign and more specifically the advertising commercials used by both candidates. It was shown in this study that ads from both sides did appeal to a lot of different emotions, even though Trump used more appeals to anger (77.3% of ads vs 53.1% for Clinton) as he was running against a candidate closely tied to the current administration in 2016. We will examine later in our paper the impact political affiliation has on the emotions of the candidates -e.g. comparing the emotions of the potential successor from the same party as the current president versus the oppositions ones.

In this matter, [Riabinin \(2009\)](#) analyzed sentiment in the debates of Canadian Parliament members, using a "bag-of-words model", which is another way to call<sup>5</sup> the Linguistic Inquiry and Word Count model (LIWC) ([Pennebaker, Francis, & Booth, 2007](#)), in order to distinguish their ideological belonging. [Leccese and Regan \(2015\)](#) underlined that the previous study found out that Liberals tended to convey positive emotion while Conservative negative ones. It is a true assertion given the results of the previous paper, however one has to be cautious not to jump to conclusions. Indeed the data from [Riabinin \(2009\)](#) are registered debates of the Parliament from 1997 to 2000, when the leading majority and thus Canadian government were liberals. So opposition members were of course more likely to employ negative terms, on the contrary to liberals representatives supporting their government.

When dealing with sentiment analysis, a tremendous source of data is Twitter. So how can we link the use of emotion extracted from images to tweet feeds ? [Bollen, Mao, and Pepe \(2011\)](#) found out that political events -such as elections in our case- are correlated with public mood. It even extends to economical events. We could also reverse the question and wonder if emotional data from political events could also influence the public opinion and thus voting outcome. On this matter,

---

<sup>5</sup>an "inelegant way", according to [Leccese and Regan \(2015\)](#)



[Mahendiran et al. \(2014\)](#) developed a forecast for elections using Twitter-derived features. They have been able to predict election results with a margin error of 5%, with a regression model based on [O'Connor, Balasubramanyan, Routledge, and Smith \(2010\)](#) and [Bermingham and Smeaton \(2011\)](#) works, which is quite a good accuracy given the field. The former paper warns about the fact that polls are not necessarily a gold standard and that they are merely a "noisy indicators of the truth". So public opinion could also be represented by other variables. The later ([Bermingham & Smeaton, 2011](#)) ask the question of representativity: are samples -even large ones- from Twitter enough to reflect on true opinion ? Indeed volume of data speaking of a candidate has been found to be the single biggest predictive variable.

## 2.3 Emotion extraction

We will not go very thoroughly through the literature of emotion feature extraction from images, for our emotion data will be provided by an already-made algorithm. However, it is important to be able to interpret the result of this predictive emotion algorithm. On a more general scale, [Wagner, Kim, and André \(2005\)](#) used a multi-layer perceptron (MLP) among other algorithms in order to classify emotions they obtained using electromyogram, electrocardiogram, skin conductivity and respiration. It is a very efficient way of determining emotions, as they found a result as accurate as 92% of the time. Especially anger is an emotion which is recognizable almost 100% of the time using these biometrics -we can guess because it affects a lot those 4 biologic parameters.

Nevertheless, when one has no access to this kind of biometrics, but only to images for instance, which accuracy can be reached ? [Busso et al. \(2004\)](#) have been able to reach a 89% overall accuracy, using a combination of acoustic and video feature analysis. Indeed, we use more than one sense to interpret our fellow humans' feelings. When only one is used, like vision, and in a static way, accuracies are of course diminished. That's why Microsoft Emotion API "is experimental, and not always accurate" according to the firm that provides it.

### 3 Motivation

Inasmuch as there is no -or little- known literature dealing with emotion extraction from images on a political application, it would be interesting to be able to link emotions from political personalities to public opinion. Our paper is an attempt to see if there is a link between the emotions of political candidates as they are represented in the press and the public opinion about them during an election campaign. Public opinion is here represented by a set of Ipsos Sopra Steria polls during the first round campaign of the 2017 French presidential election. There are 10 different polls giving voting intentions from February 02 to April 20, 2017 (Figure 1).

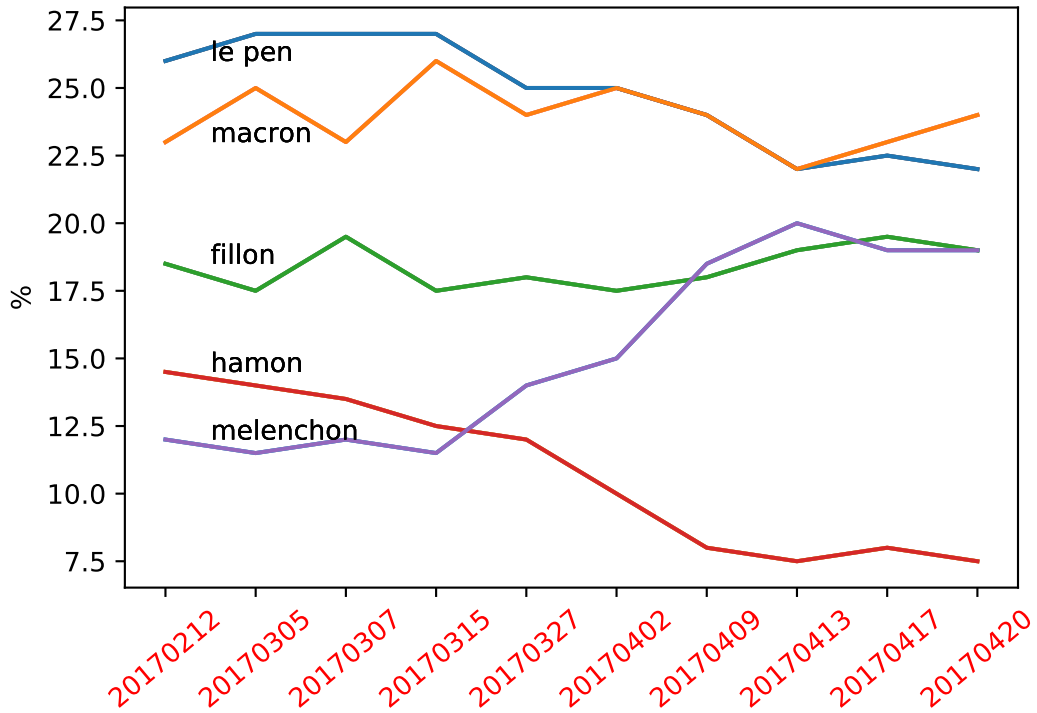


Figure 1: IPSOS SOPRA STERIA poll results

The choice of the 2017 French presidential election to build up our database is a temporal one: indeed, image "scrapping" from the web has been done just one or two month after the vote happened. Hence the still good referencing in the Google search engine of the websites -online newspaper most of the time- that provided photos of presidential candidates. Also, only the 5 best ranked candidates are here represented in order to obtain sufficient results from the Google custom search API.

The importance of well referenced websites comes from the use of the Google

Custom Search API. This API has been chosen -instead of common web scrapping- in order to specify a date range when requesting search results to be able to date the pictures downloaded. This has allowed us to download pictures from a specific date range and a specific candidate. The date range specified is the one between two poll results, allowing us to label the picture with an upward trend or a downward one, concerning voting intentions for this candidate. For more detail on the image database constitution and the use of Google Custom Search API, refer to Appendix [A](#).

The database used for the Neural Network training has been obtained using the Microsoft Emotion API as said earlier. Microsoft provide this tool, which is callable from Python and ready-to-use. On the matter of its accuracy, it not very reliable as it is still a beta-version -and known for having around 61.5% accuracy. However it's the only free tool that allows us to deal with a big quantity of data. Its output is a vector of 8 emotions : neutral, happiness, sadness, contempt, fear, surprise, anger and disgust. For more information on the Microsoft Emotion API use and the image database cleaning, refer to Appendix [B](#).

On the choice of using a Neural Network to predict poll trends using the emotion database, there are a few reasons. A modeling one is that it allows to have only this type of predictor if we also consider the first step, using the Microsoft Emotion API<sup>6</sup>, constituting all in all a "two step" Neural Network prediction algorithm. The reason is also a performance one, as we will show later.

It is indeed debatable to use emotions of candidates to predict poll trends. However, literature is scarce if nonexistent on this subject and it could be interesting to prove that there is -or there isn't- a correlation between those two concepts. As said earlier the emotions of the candidates have an effect on public opinion, especially today with the rise of fact-free politics and the use of calls to strong feelings in presidential debates ([Powell, 2016](#); [Searles & Ridout, 2017](#)). The reverse effect can also be assumed: a negative trend known by the candidate could affect his emotions. Even if he tries to control his public image, journalists that take photos in burst mode could capture the feeling they deem reflecting the candidate inner state.

---

<sup>6</sup>Even if this algorithm is a "black box" so to say, it is very likely that it uses mainly a Neural Network to predict emotions.

## 4 Descriptive Statistics

After the image data has been cleaned and converted to emotions (See Appendix B), we obtain 2151 vectors of 8 emotions corresponding each to an image. The emotion value is a percentage for each emotion, summing for each vector to 100%. The main emotion for each photo could also be an important information, so we coded two new variables : a categorical one giving the main emotion, and another one giving its percentage value.



Figure 2: Emotion distribution for all the candidates

When observing Figure 2, it is easy to recognize that two emotions are dominant in the dataset: neutral and happiness. When shifting from the total distribution (left in Figure 2) to the main emotion distribution (right in Figure 2), predictably the dominant emotions are reinforced because they are likely to have bigger values in the total distribution. On the contrary, 3 others emotions are removed from the data when considering main emotions : fear, disgust and contempt. Indeed, their weight in the total distribution were already very little, so when taking the maximum value for each picture they tend to disappear. In conclusion, the two new variables will overweight the emotion that are the most representative of the picture, and in doing so mimic human behavior which often computes only one emotion from their fellow citizens at a given instant, contrary to the emotion detection algorithm we used.

## 4.1 Emotions : a pattern for each candidate

As said earlier, the five best ranked candidates of the 2017 French presidential election -according to all the polls- have been sampled. From the political right to the left: Marine Le Pen from Le Front National<sup>7</sup>, François Fillon from Les Républicains<sup>8</sup>, Emmanuel Macron from En Marche<sup>9</sup>, Benoît Hamon from Le Parti Socialiste<sup>10</sup> and Jean-Luc Mélenchon from La France Insoumise<sup>11</sup>.

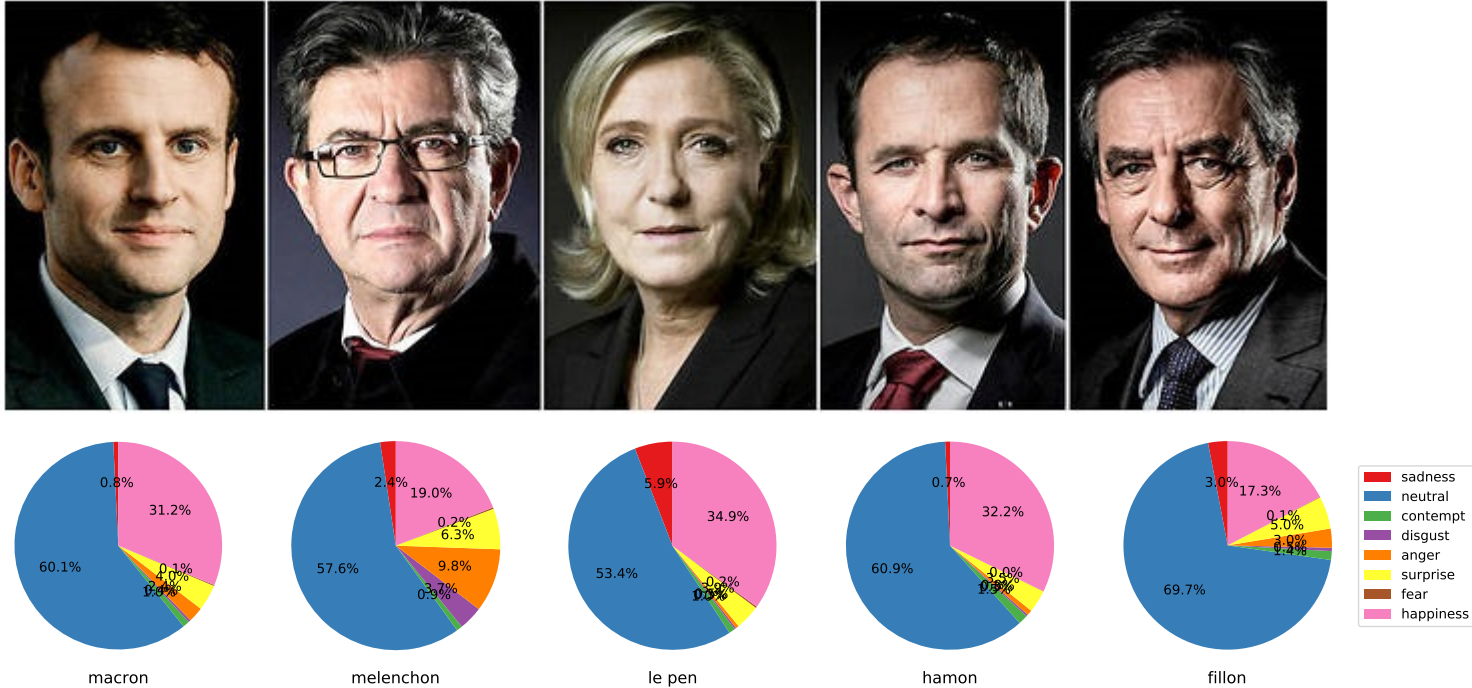


Figure 3: Comparison of the emotion distribution for each candidate

As we can observe above in Figure 3, candidates have specific emotional patterns<sup>12</sup>. The one with the bigger emotional range is Jean-Luc Mélenchon and the more neutral is François Fillon. It could be linked to their political belonging, as La France Insoumise is a revolutionary party and Les Républicains a conservative one, however it is a personal characteristic depending on the personality of the candidate first and foremost. Since the candidates seem to have a different emotional pattern, we code a categorical variable that say to which candidate belongs the face in the picture. Of course we can guess that a different candidate would have a different emotional response to events such as a shift of trend in the polls.

<sup>7</sup>For more info on Le Front National: <http://www.frontnational.com/>

<sup>8</sup>For more info on Les Républicains: <https://www.republicains.fr/>

<sup>9</sup>For more info on La République en Marche: <https://en-marche.fr/>

<sup>10</sup>For more info on Le Parti Socialiste: <http://www.parti-socialiste.fr/>

<sup>11</sup>For more info on La France Insoumise: <https://lafranceinsoumise.fr/>

<sup>12</sup>Refer to Appendix C for the main emotion comparison between the candidates and more.

## 4.2 Difference between upward trends and downward trends

From the search in the Google Custom Search API (See Appendix A), we have been able to identify the candidate and the period of time for each image, in order to specify a poll trend<sup>13</sup>. All in all, 904 photos have been published when the candidate was in an upward trend, 194 when stagnating in the polls, and 1053 in a downward trend. In order to simplify the problem in a binary way, we chose to count stagnation as an upward trend. Indeed the candidate that keep his result in the polls could see it as a positive result as he is not losing any votes<sup>14</sup> -and already part of the 5 best candidates. The contrary may also be arguable -that stagnation is negative-, however the former seemed to be more likely given politics nowadays.



Figure 4: Comparison of emotion distributions between upward trends and downward trends

On the graph above (Figure 4), we can observe that the total distribution does not variate a lot over the different trends. Candidates are more neutral (+2.1 pt) and less happy (-4.2 pt) when their polls have a negative slope rather than positive. However, on the graph below (Figure 5), speaking of main emotions, the shift in

<sup>13</sup>The first trend was deducted from another IPSOS SOPRA STERIA poll dating back from January 15, 2017, in order to have 11 poll dates and 10 trend periods.

<sup>14</sup>The presence of scandals in the campaign showed that candidates are likely to lose a lot of voting intentions over them. For instance François Fillon lost almost 7% in a few weeks over an alleged embezzlement accusation.



neutrality is a little bigger (+3.2 pt), but the negative shift in happiness stays the same.

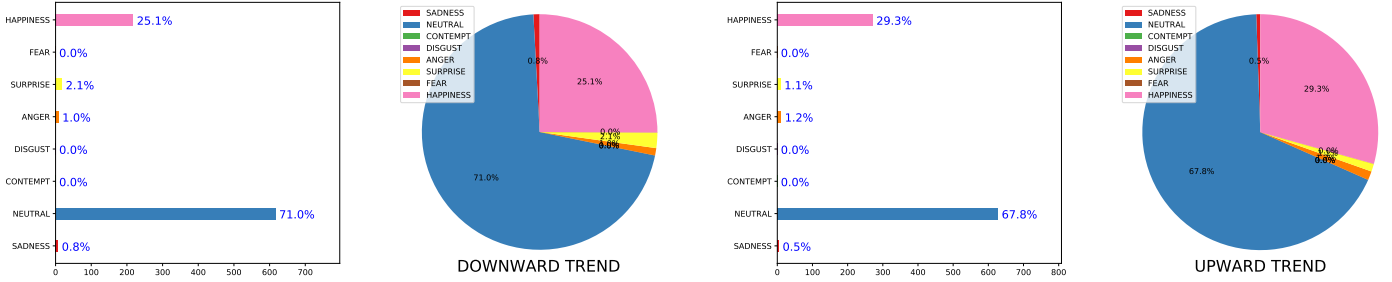


Figure 5: Comparison of main emotion distributions between upward trends and downward trends

The critical question is now whether this difference between upward and downward trend emotions will be enough to classify the images. Of course a descriptive overview is not enough to decide if it is enough to separate between the two. For instance, a non-linear pattern could be found by a non-linear algorithm, such as a Multi Layer Perceptron.

## 5 Data Preparation

The problem we have chosen to analyze is the possibility of predicting poll trends based on the emotions of the candidates. A Neural Network, and more specifically a Multi Layer Perceptron (MLP) has been used to do so. Nevertheless, data needs to be prepared first and foremost. Neural Networks don't need *ex ante* feature selection, however certain form of data is required in order to achieve sufficient performance. The database we have at this point consists of 11 variables : 8 emotions scores, the main emotion category, its value and the candidate category. Plus the output variable which is the trend of the corresponding poll : -1 for the downward trend and +1 for the upward (and stagnation).

As far as categorical variable are concerned, [Sarle \(2002\)](#) and [Lessmann \(2017\)](#) indicate that it's necessary to transform it to replicate those variables into as much dummy<sup>15</sup> variables needed. More specifically, if we have N categories, it will be transformed to N-1 dummy variables in order to avoid linear dependency. However,

<sup>15</sup>also called one-hot encoding in machine learning.

both previous authors agree that it is also better to code dummy variables in a  $\{+1,-1\}$  binary way, in order to let the initial decision hyperplanes to cut the data in a variety of directions and thus finding more local optima. We finally have 17 different variables : 8 emotions values, 1 main emotion values, 4 binary variables for the 5 main emotions that remain - 3 were never main emotions as seen in the descriptive statistics - and one was omitted to avoid linear dependency<sup>16</sup>, and 4 binary variables for the 5 candidates<sup>17</sup>. Refer to Table 6, Appendix D for more details on the variables and their treatment.

Dealing now with numerical variables, we perform a Principal Component Analysis (PCA) over those. This allow us to observe data on a two-dimensional plane to be able to observe their spread across a reduced-dimension space.

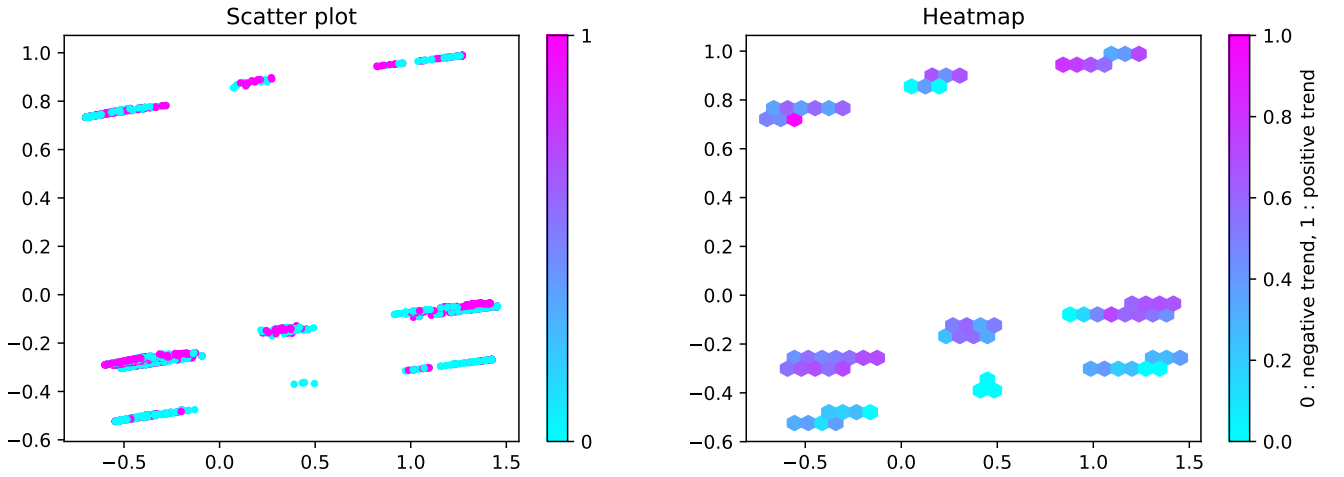


Figure 6: PCA of data before standardization

On Figure 6 one can observe a 2D scatter plot and heat map of the data along the first two eigenvectors of the PCA. Indeed, data has not yet been standardized or normalized, explaining the fact that it is not very scattered. As said *supra*, it is better to have data centered at the origin for initialization, so standardization will be chosen for the data<sup>18</sup>. Figure 7 is a 2 component<sup>19</sup> PCA applied to the data after it has been standardized, e.g. with a distribution of mean 0 and variance

<sup>16</sup>*neutral\_main* was dropped for the main emotion categorical variables.

<sup>17</sup>*macron* was dropped for the candidate categorical variables.

<sup>18</sup>It also avoids eliminating one emotion value variable, because they were supposed to be summing to 100% (which they didn't exactly anyway, after checking the output of the Microsoft Algorithm, probably because of rounding) to avoid linear dependency.

<sup>19</sup>The explained variance ratios were 0.24, 0.18 and 0.09 for the 3 first eigenvectors on a 3 component PCA, so the 3D representation along a third axis wouldn't have brought much more information. It was confirmed with 3D plotting of this 3 component PCA.



1. We can see that the data is then better scattered across space, indicating that decision making will be easier for algorithms that use it as input. For the binary categorical variables, we do not need to standardize them because they have already been encoded as  $\{+1,-1\}$  binary variables (Sarle, 2002).

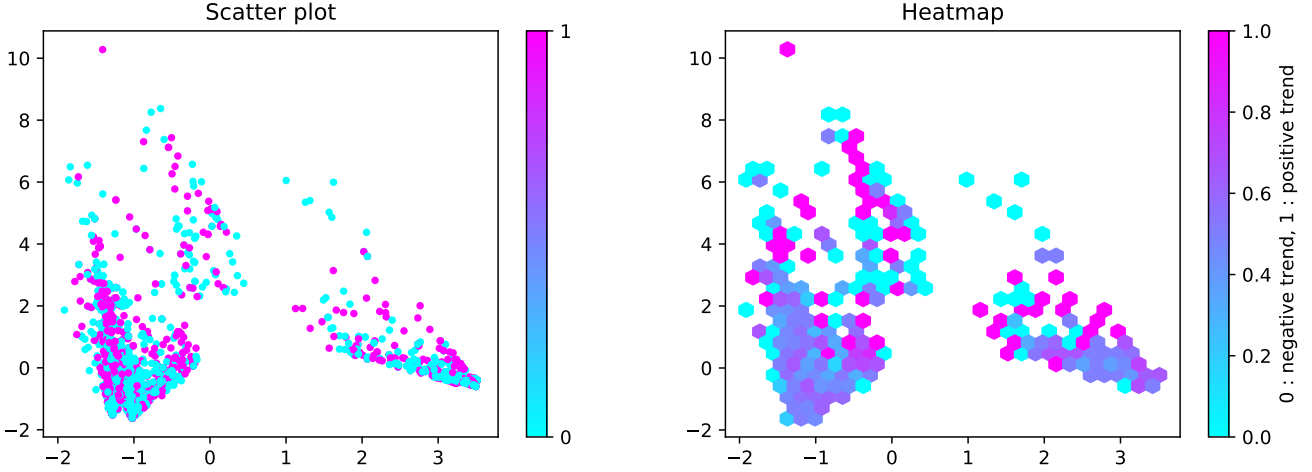


Figure 7: PCA of data after standardization

Finally, as our dataset is slightly unbalanced - 1053 photos were published while in a downward trend, while 1098 in an upward trend<sup>20</sup> - a balanced dataset has been created. It has been done so using the class weighting inspired by Logistic Regression in Rare Events Data from King and Zeng (2001).

As far as outliers are concerned, Sarle (2002) again says that irrelevant inputs are close to be ignored by MLP, so no outlier selection will be here performed. Furthermore, the way our database was created (see Appendix B), with a control which was almost manual, allows us to say that every image should be part of the dataset.

## 6 Model Creation & Optimization

For our prediction of the poll trends, we used a Neural Network MLP. In order to predict the target data, which is for now the sign of the slope of the poll curve during the period considered, we need a dummy Y variable, instead of a  $\{+1,-1\}$  binary one. Indeed the -1 (downward trend) has been transformed to 0 for the needs of the output layer of the MLP, which has *softmax* as an activation function.

<sup>20</sup>From now on, for simplification purposes, upward trend signifies upward or stagnation.

## 6.1 Model Creation

The MLP has been implemented using the module Keras in Python (for programming details refer to Appendix E). First of all, our Neural Network is consisting of an input layer, hidden layer(s) and an output layer.

In order to avoid overfitting, as advised by Lessmann (2017), regularization has been introduced using L2 Gaussian penalty. When doing so the accuracy of our algorithm is also increased, confirming the theory that before regularization it could overfit the training sample and is less generalizable to other samples.

As far as the data transmission inside the network is concerned, some batch normalization help preparing the data output from a layer into feeding it as input for the next layer (Ioffe & Szegedy, 2015). Indeed, for the first layer we have prepared the data by standardizing it, however the weighted data out of a specific layer is no longer well distributed. Batch normalization then help reduce the shift in covariance and thus stabilize and quickens the learning.

We chose to leave (Appendix B) the redundant image data in the dataset to be still representative of the press that sometimes use the same pictures as other websites in their articles. It could be argued that redundancy affects the algorithm, however Sarle (2002) argues that "redundant inputs has little effect on the effective dimensionality of the data" and that MLP process it effectively.

Even with the use of regularizers, we can observe that overfitting could be still be a problem for our algorithm. In order to eliminate this effect, neuron dropout have been added in between layers. Dropout is an algorithm for training neural networks which relies on stochastically "dropping out" neurons during training in order to avoid overfitting (Baldi & Sadowski, 2013). During each epoch, a random fraction of dropout neurons are not used, thus reducing interdependent learning amongst the neurons and overfitting.

## 6.2 Model Selection and Parametrization

First of all, the data has been split into 3 parts : train set (70%), validation set<sup>21</sup> (15%) and test set (15%). In order to evaluate the model, accuracy and ROC

---

<sup>21</sup>In this paper, understand validation in an NN way, the sample validating the neuron weights every epoch.

AUC<sup>22</sup> on the test set have been used, taking into account that some callbacks have been implemented in order to have a good AUC while training. The loss function paired with the *softmax* output layer activation function is the *sparse categorical crossentropy* which allows the output label to be a single dummy variable for the binary classification problem we consider.

After defining the general shape of our MLP, at this point the critical question is now how to be able to chose all its parameters. The module Hyperopt (Hyperas when paired with Keras) has been of great use in doing so. It allowed to run the model fitting multiple times and then compare the results of accuracy and ROC AUC (indirectly, see *supra* and Appendix E) and then chose the best model.

All the parameters that were tested are the following : the number of hidden layers (1 or 2), the number of neurons in each layer, the dropout frequency, which activation function to choose (except the last one, which has been already set to *softmax*), the optimizer we use for finding the optima, its learning rate. In addition, we also tune the batch size to compromise with the learning rate. The result chosen for our model selection is in the following Table 1:

Layer	Parameter	Parameter value
	Number of hidden layers	2
Input	Number of neurons	32
Input	Activation function	relu
	Dropout Frequency	0.41
1	Number of neurons	64
1	Activation function	sigmoid
	Dropout Frequency	0.09
2	Number of neurons	32
2	Activation function	selu
	Dropout Frequency	0.21
Optimizer	Function	Adamax
Optimizer	Learning rate	0.0005
Fitting	Batch size	256
Fitting	Epoch	600

Table 1: Resulting parameters from the optimizations

For more information on the shape of the MLP, refer to Figure 17 in Appendix

<sup>22</sup>Receiver operating characteristic - area under curve

F. The results of this Neural Network will be presented in the next section.

## 7 Empirical Results

We now present the results by the optimized MLP on the 17 input variables. The trainings have been implemented 10 times in order to have more accurate results, as seen on Table 2. For more information on the training curves of the MLP refer to Appendix G (Figure 18 and Figure 19). So far, we have been able to achieve an accuracy around 64%. The ROC AUC callback has helped us achieve also an AUC around 63%.

Value	Accuracy		Categorical crossentropy		ROC AUC
	Validation	Test	Validation	Test	Test
Mean	64.1%	63.8%	0.698	0.698	63.0%
Standard deviation	1.4%	1.0%	0.007	0.008	0.9%

Table 2: Results of the initial MLP on the 17 variables.

The predicted classes can be represented in a confusion matrix on the test set to see how the algorithm behaves regarding prediction, as showed on Figure 8. It represent the prediction results for one of the trainings of the Table 2.

Thus the prediction results tend to show us that it is easier predicting upward trend in the polls based on the emotions of the candidates. The conclusions we can draw from this aspect will be discussed later. However, the precision we have reached implicates that emotions alone may not be enough to predict the outcome of a poll. It may give however an idea of the correlation between poll and emotion time series.

In order to compare the performance of our MLP with another algorithm, we have performed a Light Gradient Boosting using the package LightGBM. The result gives 60.6% accuracy, however the confusion matrix is even more shifted towards the upward trend. Kuhn and Johnson (2013) imply that gradient boosting is vulnerable to overfitting, which could be the case in comparison with our MLP model, given all the regularizers and neuron dropouts we used.

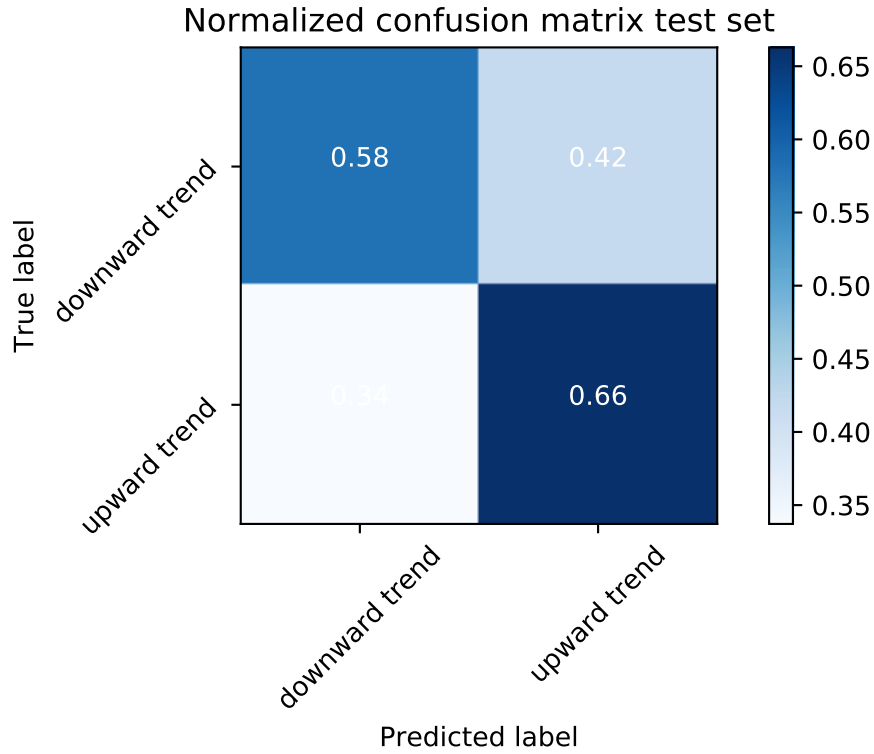


Figure 8: Confusion matrix of the MLP on 17 variables

## 7.1 Variable importance

With the purpose to simplify the model while achieving the same accuracy, we have reduced the number of variables using a wrapper strategy. [Lessmann \(2017\)](#) indicates to remove each variable one by one and run the MLP. The removed variable -giving best AUC when the MLP is fitted without it- is the *neutral\_value* variable. We compute the second step of the wrapper and the removed variable is *happiness\_value*. On the third step, accuracy sharply decreases so we stop the wrapper.

On Table 3, we can observe the result of the first step of the wrapper. For better precision the MLP has been trained and tested 10 times on each variable removal.

There are at the end 15 variables left on our dataset, while *neutral\_value* and *happiness\_value* have been removed. It is interesting to note that those were the two emotions that had the biggest values and were the most represented for the candidates.

Removed Variable	Accuracy		ROC AUC
	Validation	Test	Test
sadness_value	63.4%	63.9%	63.3%
neutral_value	63.9%	64.3%	63.4%
contempt_value	65.0%	63.4%	63.0%
disgust_value	64.2%	62.1%	61.5%
anger_value	64.2%	63.2%	62.4%
surprise_value	63.7%	63.4%	62.6%
fear_value	63.4%	63.1%	62.1%
happiness_value	64.1%	64.1%	63.3%
main_value	65.2%	63.7%	63.0%
sadness_main	64.0%	63.0%	62.5%
anger_main	63.9%	63.7%	63.0%
surprise_main	63.6%	62.8%	62.1%
happiness_main	64.8%	63.6%	62.7%
lepen	64.5%	63.6%	62.8%
fillon	61.0%	63.6%	62.1%
hamon	55.8%	55.3%	56.1%
melenchon	63.7%	63.7%	62.9%

Table 3: Results of the first step of the wrapper

## 7.2 Presentation of final results

Table 4 displays the final results of our study, a 2 hidden layers Perceptron with 15 dimensions input, predicting a binary dummy output.

Value	Accuracy		Categorical crossentropy		ROC AUC
	Validation	Test	Validation	Test	Test
Mean	63.5%	64.2%	0.700	0.696	63.5%
Standard deviation	0.9%	0.9%	0.007	0.007	1.2%

Table 4: Results of the final MLP on the 15 selected variables.

Finally, our MLP applied to 15 variables allows an accuracy around 64%. That does mean that if we feed the image of one of the five candidates to the Microsoft Emotion API and then to our MLP, we could say with 64% confidence that the candidate is in an upward or a downward trend. As far as variable interpretation is concerned, the Neural Network is a tool that does not allow it. However it could be used with specific emotions in photos to see which emotion distribution implies which trend.

The ROC curve of the MLP (Figure 9) when predicting on the test set shows that the callback function also optimized the AUC. Indeed usually the ROC curve is much more less straight. When not using a callback the curve is more stochastic. We can deduce from this curve that the power of our classifier is intermediate.

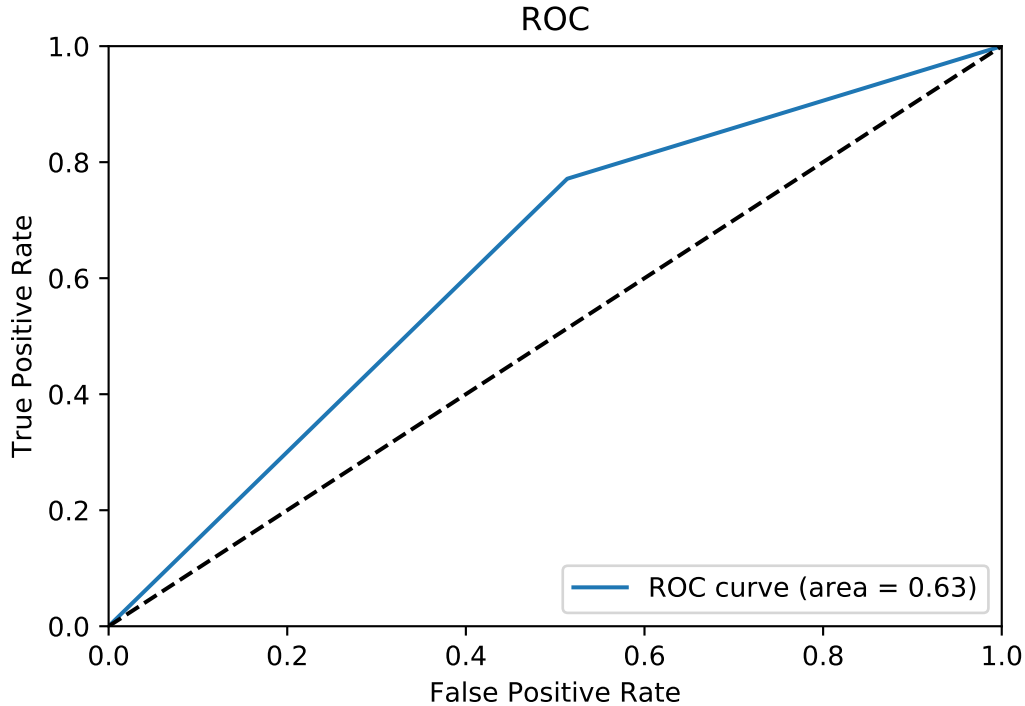


Figure 9: ROC curve of the MLP on 15 variables

## 8 Discussion

The objective of this paper was to demonstrate whether there was a link between the emotion of politicians and the voting intention trends by trying to predict the trends with the emotional data. The result we achieved so far seems to show that there is a link since it has been possible to predict the poll slope sign more than 50% of the time. However, the reached accuracy (64%) seem to be not so performing given what can be achieved using sentiment analysis. It is also useful to see that it's easier to predict the upward trend rather than downward ones. One can assume that maybe patterns are more recognizable when the candidate is in a positive state than when in a negative state, leading to bursts of non usual emotions.

Nevertheless, it is important to specify that facial emotion are not entirely corre-

lated with poll trends, so the result we have can give an idea of the correlation that exists between facial emotions and poll results. Those two concept seem to interact in the way that when voters see the pictures of politicians in the press, and especially in our case their emotions, it could influence partially their way of assessing the candidate. Conversely, poll trends could also affect the candidate reactions and facial emotions, depending on whether his pollsters -or the official polls- say that he rises or goes down in the polls.

As far as generalization is concerned, we cannot draw conclusions from this study on an universal basis. [Mahendiran et al. \(2014\)](#) argues that when using a non representative sample, it implies bias in following election dynamics. Indeed, our data is a choice among millions of pictures that were taken during the campaign and sampling has been limited because of the computational limits of the Google Custom Search API. Furthermore, the Microsoft Emotion API is still in experimental phase, and waiting for the final version to be released could improve our results.

## 9 Conclusion

The link between emotions of politicians in the press and voting intentions seem finally to be moderate in our sample, however we cannot generalize to other samples. An extensive research on many more samples should be conducted in order to draw a viable and robust conclusion.

This framework could open possibilities on applying this model to consulting for politicians. Indeed if we feed a few photos of a candidate for an election to the MLP, we could classify the faces the candidate should make and those he shouldn't, given the results of previous poll trends. Emotional intelligence advisory indeed becomes more and more popular in firms and also politics, so this model could provide complimentary and additional information on indicated behavior for candidates.

Further research on this topic could also be made dealing with times series of emotions and polls. This would allow the computation of cointegration between emotions on pictures and poll results. However this would imply having very precise dates for a lot of image data, which seem very hard to achieve given our data. Hence the importance of gathering more data on politicians pictures.



Instead of the sign of the poll slope, the sign of poll acceleration could also have been studied. Indeed, the candidate could have different reactions when he is on an upward trend since a long time, compared to when he has been in a downward trend just before. This also would require a lot more data since trend breaks are not so common, especially in such a short time range as a presidential election campaign.

## References

- Baldi, P., & Sadowski, P. J. (2013). Understanding dropout. In *Advances in neural information processing systems* (pp. 2814–2822). Retrieved from <http://papers.nips.cc/paper/4878-understanding-dropout.pdf>
- Bermingham, A., & Smeaton, A. (2011). On using twitter to monitor political sentiment and predict election results. In *Proceedings of the workshop on sentiment analysis where ai meets psychology (saaip 2011)* (pp. 2–10). Retrieved from <http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.222.2863&rep=rep1&type=pdf>
- Bollen, J., Mao, H., & Pepe, A. (2011). Modeling public mood and emotion: Twitter sentiment and socio-economic phenomena. *International AAAI Conference on Weblogs and Social Media*, 11, 450–453. Retrieved from <https://www.aaai.org/ocs/index.php/ICWSM/ICWSM11/paper/viewFile/2826/3237>
- Busso, C., Deng, Z., Yildirim, S., Bulut, M., Lee, C. M., Kazemzadeh, A., ... Narayanan, S. (2004). Analysis of emotion recognition using facial expressions, speech and multimodal information. In *Proceedings of the 6th international conference on multimodal interfaces* (pp. 205–211). Retrieved from [http://www.cvauni.edu.vn/imgupload\\_dinhkem/file/Chuyen%20De%20HCI/3\\_%20Analysis%20of%20Emotion%20Recognition%20using%20facial%20expression-speech.pdf](http://www.cvauni.edu.vn/imgupload_dinhkem/file/Chuyen%20De%20HCI/3_%20Analysis%20of%20Emotion%20Recognition%20using%20facial%20expression-speech.pdf)
- Ioffe, S., & Szegedy, C. (2015). Batch normalization: Accelerating deep network training by reducing internal covariate shift. In *International conference on machine learning* (pp. 448–456). Retrieved from <https://arxiv.org/pdf/1502.03167.pdf>
- King, G., & Zeng, L. (2001). Logistic regression in rare events data. *Political analysis*, 9(2), 137–163. Retrieved from <https://dash.harvard.edu/bitstream/handle/1/4125045/relogit%20rare%20events.pdf?s>
- Kuhn, M., & Johnson, K. (2013). *Applied predictive modeling* (Vol. 26). Springer.
- Leccese, M., & Regan, M. M. (2015). Angry words: A content analysis of emotion in political blogs and op-ed columns. *Online Journal of Communication and Media Technologies*, 36(3), 27–36. Retrieved from <http://www.ojcmt.net/articles/53/532.pdf>

- Lessmann, S. (2017). *Business analytics and data science*. University Lecture, Humboldt University.
- Mahendiran, A., Wang, W., Lira, J. A. S., Huang, B., Getoor, L., Mares, D., & Ramakrishnan, N. (2014). Discovering evolving political vocabulary in social media. In *Behavior, economic and social computing (besc), 2014 international conference on* (pp. 1–7). Retrieved from [http://people.cs.vt.edu/naren/papers/election\\_besc14\\_final.pdf](http://people.cs.vt.edu/naren/papers/election_besc14_final.pdf)
- O'Connor, B., Balasubramanyan, R., Routledge, B. R., & Smith, N. A. (2010). From tweets to polls: Linking text sentiment to public opinion time series. *ICWSM*, 11(122-129), 1–2. Retrieved from <https://homes.cs.washington.edu/~nasmith/papers/oconnor+balasubramanyan+routledge+smith.icwsml0.pdf>
- Pennebaker, J. W., Francis, M. E., & Booth, R. J. (2007). Linguistic inquiry and word count: Liwc 2007. *Mahway: Lawrence Erlbaum Associates*, 71. Retrieved from <http://www.depts.ttu.edu/psy/lusi/files/LIWCmanual.pdf>
- Powell, A. (2016). Algorithms, accountability, and political emotion: on the cultural assumptions underpinning sentiment analysis. *LSE Impact of Social Sciences Blog*. Retrieved from <http://blogs.lse.ac.uk/impactofsocialsciences/2016/07/20/algorithms-accountability-and-political-emotion/>
- Riabinin, Y. (2009). Computational identification of ideology in text: A study of canadian parliamentary debates. *MSc paper, Department of Computer Science, University of Toronto*. Retrieved from <http://ftp.cs.toronto.edu/pub/gh/Riabinin-MSc-paper.pdf>
- Sarle, W. S. (2002). *Faq on neural networks. part 2: Learning*. Retrieved from <http://www.faqs.org/faqs/ai-faq/neural-nets/part2/>
- Scheff, T. J. (2003). Shame in self and society. *Symbolic interaction*, 26(2), 239–262.
- Scheff, T. J. (2005). *Emotions and politics*. Retrieved from [https://www.youtube.com/watch?v=iw\\_6ayN00q8](https://www.youtube.com/watch?v=iw_6ayN00q8) (Emeritus professor of Sociology at UCSB, an expert on the sociology of emotions, discusses his research on male emotions and violence, as well as his thoughts on the role of shame and alienation in destructive conflict.)
- Searles, K., & Ridout, T. N. (2017). The use and consequences of emotions in pol-

itics. *Emotion Researcher*. ISRE's Sourcebook for Research on Emotion and Affect, Andrea Scarantino (Ed.). Retrieved from <http://emotionresearcher.com/the-use-and-consequences-of-emotions-in-politics/>

Wagner, J., Kim, J., & André, E. (2005). From physiological signals to emotions: Implementing and comparing selected methods for feature extraction and classification. In *Multimedia and expo, 2005. icme 2005. ieee international conference on* (pp. 940–943). Retrieved from <http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.470.9227&rep=rep1&type=pdf>

# Appendix

## A Image Database creation

As we use 11 different poll results (including the initial one from mid-January 2017) for 5 candidates, we have 10 poll periods in between. The difficulty was to find images from the web which had the date information along with the candidate label. Using normal web scrapping would have raised an issue concerning those two aspects, and especially the date one.

### A.1 The Google Custom Search API

When searching in the Google search bar a sentence, like the one we used for instance "photos discours macron presidentielles 2017" which translate to "pictures speech macron presidential election 2017", the result that are presented to us in Google are mainly articles from online newspapers. They appear to have a date of publication at the beginning of the short description Google provides us. However, it would have been too long to manually download thousands of pictures.

So, in order to create the image database, we used the Google Custom Search API. This API is available through the Google developer network. It allows to create automatic queries from the Google Search engine. The number of queries per minute on the free version were restrictive so we created a billing account in order to get enough data download frequency. The interface we use to implement it was Python.

### A.2 Getting the initial image database

As said earlier, the Google search engine provides a release date for articles. But the Google Custom Search (GCS) API does not provide any sort of date in its result. The way we dealt with this issue is the following: after research, an option in the query of GCS is to specify a certain date range, so we put the date range between two polls into the query. Along with the package urllib2, we have been able to make 5000 queries (100<sup>23</sup> for each of the 5 candidates at each 10 poll periods).

---

<sup>23</sup>100 is approximatively the number of results that are still relevant regarding the search query.

## B Data Cleaning & Emotion Recognition

### B.1 Necessity of a "manual" cleaning

When the initial image data has been downloaded, the image resulting of the GCS had a specific candidate and date label. However, because it was uploaded from the Google Search engine, the image could show a different candidate or several of them. Or even a complete different person (for instance a family member of the candidate, one of his political supporters ...). In order to deal with such an issue, we have performed a manual cleaning by going through all of the database. Indeed, an image recognition algorithm could have been quicker, however its precision would have forced us to put a threshold to keep only the well labeled picture and thus "losing" more data. Hence our time consuming choice of manual cleaning to keep the maximum image data.

### B.2 The Microsoft Emotion API

The Microsoft Emotion API (ME) is part of the Microsoft Cognitive Services, tools that allow the user to recognize faces, speeches or emotions. As for the Google Custom Search API, the Emotion API has a restrictive frequency limitation, so a billing account has been set in the Microsoft Azure Cloud to prevent queries limitation.

```
[{'u'faceRectangle': {'u'width': 70, 'u'top': 101, 'u'left': 97, 'u'height': 70}, 'u'scores': {'u'sadness': 0.00120383676, 'u'neutral': 0.6204358, 'u'contempt': 0.00388511736, 'u'disgust': 0.0004799715, 'u'anger': 0.000293028774, 'u'surprise': 5.940752e-05, 'u'fear': 2.6531352e-06, 'u'happiness': 0.3736402}}]
```



Figure 10: Example of output from the Microsoft Emotion API

ME provides a result with the place of the face detected and the percentage result of the emotions of this face. The 8 emotions detected are : sadness, neutral, contempt, disgust, anger, surprise, fear and contempt. When several faces are detected, the result is given first for the bigger face, then for the second bigger, etc. This aspect has been useful for the manual cleaning. Indeed when one of the face was obviously bigger, we kept the image if it was the one of the candidate. If there was a doubt (which was the case quite a few times), we ran the image through the ME to see which face was bigger. If the candidate’s face was the bigger, we kept the image. We removed it otherwise because it would have taken a tremendous amount of time -in amount of the already very time consuming manual cleaning- to label each of the 4369 images with the rank of the candidate’s face.

On Figure 10 we can observe the result of the Microsoft Emotion API on one of the images of the database. A visual rectangle around the face have been implemented using the module cv2 from python in order to make the manual cleaning easier. On the image we also wrote the main emotion of the face detected.

### **B.3 Final image database & emotion database creation**

Table 5 on next page displays the images downloaded via Google Custom Search (Initial column) and the images kept after cleaning (Cleaned column). The Initial column doesn’t contain 100 pictures for each date and candidate, because url downloading has not always been possible.

The redundant images have been kept in our database. Indeed some articles use the same pictures as others, or two Google references could lead to the same article. To be representative of the reality, we chose to keep duplicates, as it is more probable that a reader we see this particular picture when going through the news. Moreover the MLP is able to process redundancy, as said in section 6.1.

After the images were run through ME, we obtained 2151 emotion vectors. Indeed Microsoft Emotion API requires a certain sharpness of the face in order to detect its place and its emotions, leading to loss of data.

Date range	Fillon		Hamon		Le Pen		Macron		Melenchon	
	Initial	Cleaned	Initial	Cleaned	Initial	Cleaned	Initial	Cleaned	Initial	Cleaned
02/00-02/12	94	69	94	59	92	62	92	58	92	50
02/12-03/05	93	77	97	59	89	55	91	62	94	48
05/05-03/07	94	77	55	24	31	15	18	16	24	12
03/07-03/15	87	63	94	58	95	53	85	53	92	49
03/15-03/27	94	62	94	55	92	45	94	48	97	55
03/27-04/02	90	45	90	51	93	54	96	60	99	63
04/02-04/09	95	56	93	29	93	47	94	42	97	54
04/09-04/13	91	61	89	52	92	48	85	53	95	48
04/13-04/17	94	72	94	63	81	48	95	50	93	54
04/17-04/20	93	55	95	69	88	62	84	32	86	43
Total	925	637	895	519	846	489	834	464	869	476
Total of Initial 4369				Total of Cleaned 2585						

Table 5: Table of images downloaded and remaining after manual cleaning



## C More on Descriptive Statistics

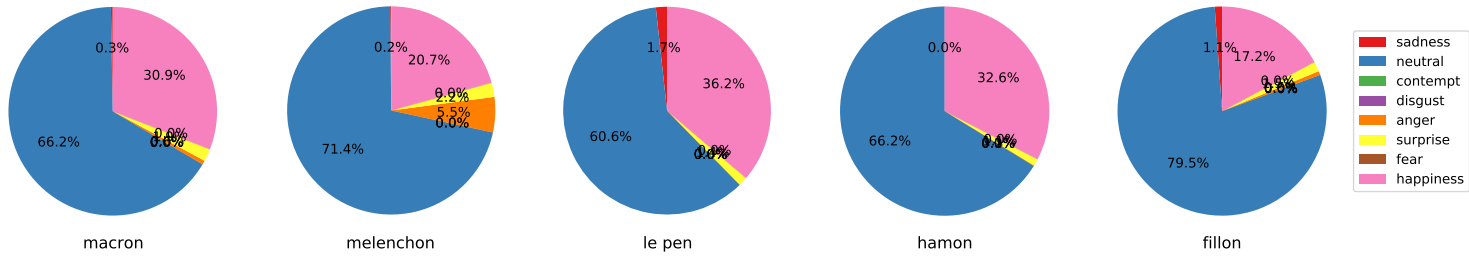


Figure 11: Comparison of the main emotion distribution for each candidate

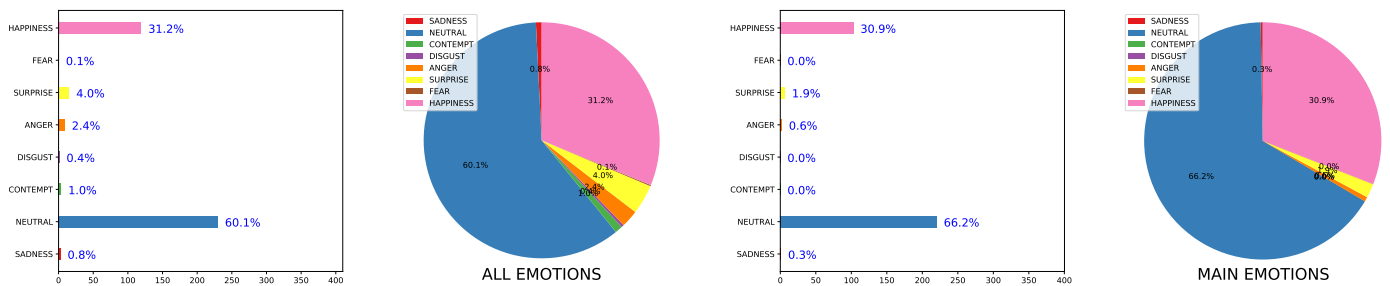


Figure 12: Emotion distribution for Emmanuel Macron

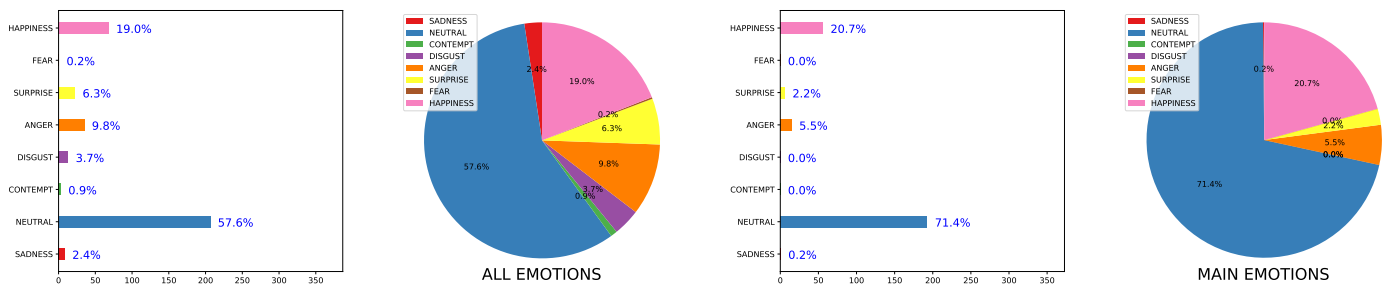


Figure 13: Emotion distribution for Jean-Luc Mélenchon

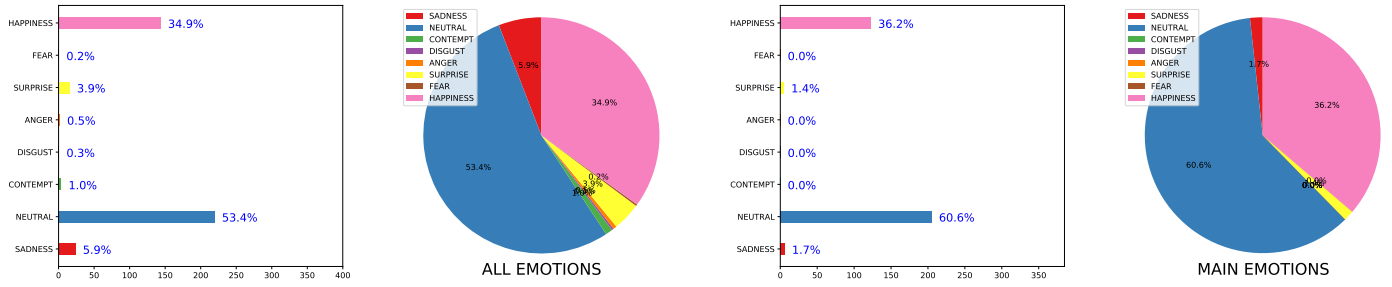


Figure 14: Emotion distribution for Marine Le Pen

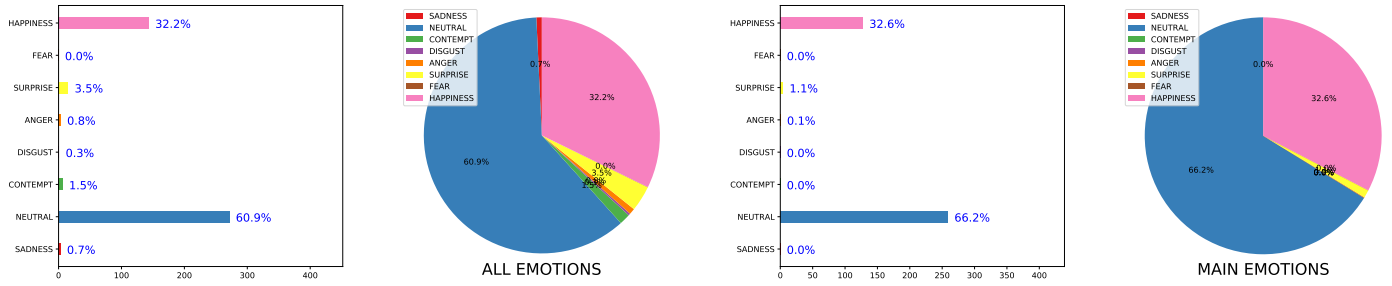


Figure 15: Emotion distribution for Benoît Hamon

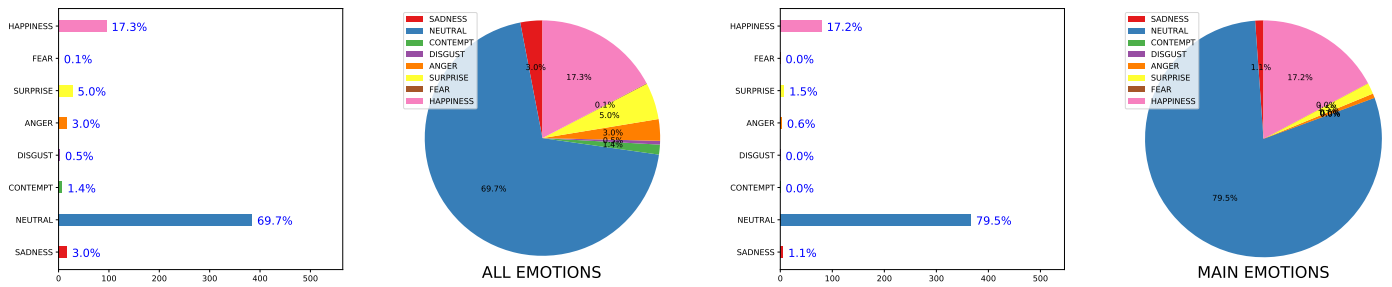


Figure 16: Emotion distribution for François Fillon

## D Table of Variables

Variable Number	Variable Name	Original Type	Treatment	Final Type
1	sadness_value	float	standardization	float
2	neutral_value	float	standardization	removed**
3	contempt_value	float	standardization	float
4	disgust_value	float	standardization	float
5	anger_value	float	standardization	float
6	surprise_value	float	standardization	float
7	fear_value	float	standardization	float
8	happiness_value	float	standardization	removed**
9	main_value	float	standardization	float
10	sadness_main	categorical	one-hot	+1/-1
11	neutral_main	categorical	one-hot	removed*
12	anger_main	categorical	one-hot	+1/-1
13	surprise_main	categorical	one-hot	+1/-1
14	happiness_main	categorical	one-hot	+1/-1
15	lepen	categorical	one-hot	+1/-1
16	fillon	categorical	one-hot	+1/-1
17	macron	categorical	one-hot	removed*
18	hamon	categorical	one-hot	+1/-1
19	melenchon	categorical	one-hot	+1/-1
20	poll_trend	int	re-categorization	dummy

Table 6: Table of initial variables and treatment applied

\*to avoid linear dependancy

\*\*to simplify the model when checking variable importance

## E Neural Network Programming Details

We have implemented the Multi Layer Perceptron using the Keras module from Python, and following have optimized its parameters using the Hyperopt module. Other modules have been used in order to produce graphics and result interpretations, such as matplotlib, sklearn, pydot or graphviz.

### E.1 The Keras Module

In order to install Keras on the computer, it is necessary to remove all Python versions in order to follow very specific installation steps. Among others, Tensorflow must be installed before Keras. Indeed, the Keras module is built on top of the Tensorflow API and use it as a basis to build Neural Networks. Keras provides then a simplified interface with the user, using Model and Sequential APIs to build a Multi Layer Perceptron for instance.

All the elements introduced in section 6.1 can be built using these APIs: input, output and hidden layers, dropout layers, batch normalization, regularization, activation functions. One can observe them in Appendix F, Figure 17. The model is then compiled, specifying a loss function - *sparse categorical crossentropy* as far as we are concerned - and an optimizer.

Before fitting the model, we have built a callback function that allows to control the ROC AUC score during the fitting of the MLP. This makes possible to fit the model using a loss function and accuracy as a metrics, while controlling for AUC each epoch. In Keras, the fitting specifies how many epoch you need and the batch size -the number of samples that will be propagated through the network.

### E.2 The Hyperopt Module

The Hyperopt module, when paired with the Keras module, is named Hyperas. It allows to run the model a specified amount of times with different parameters and chose the best model, based on its accuracy. Given the number of parameters to tune, it has been used many times in order to first find approximate values of all parameters, and then becoming more and more accurate.

## F Multi Layer Perceptron Model

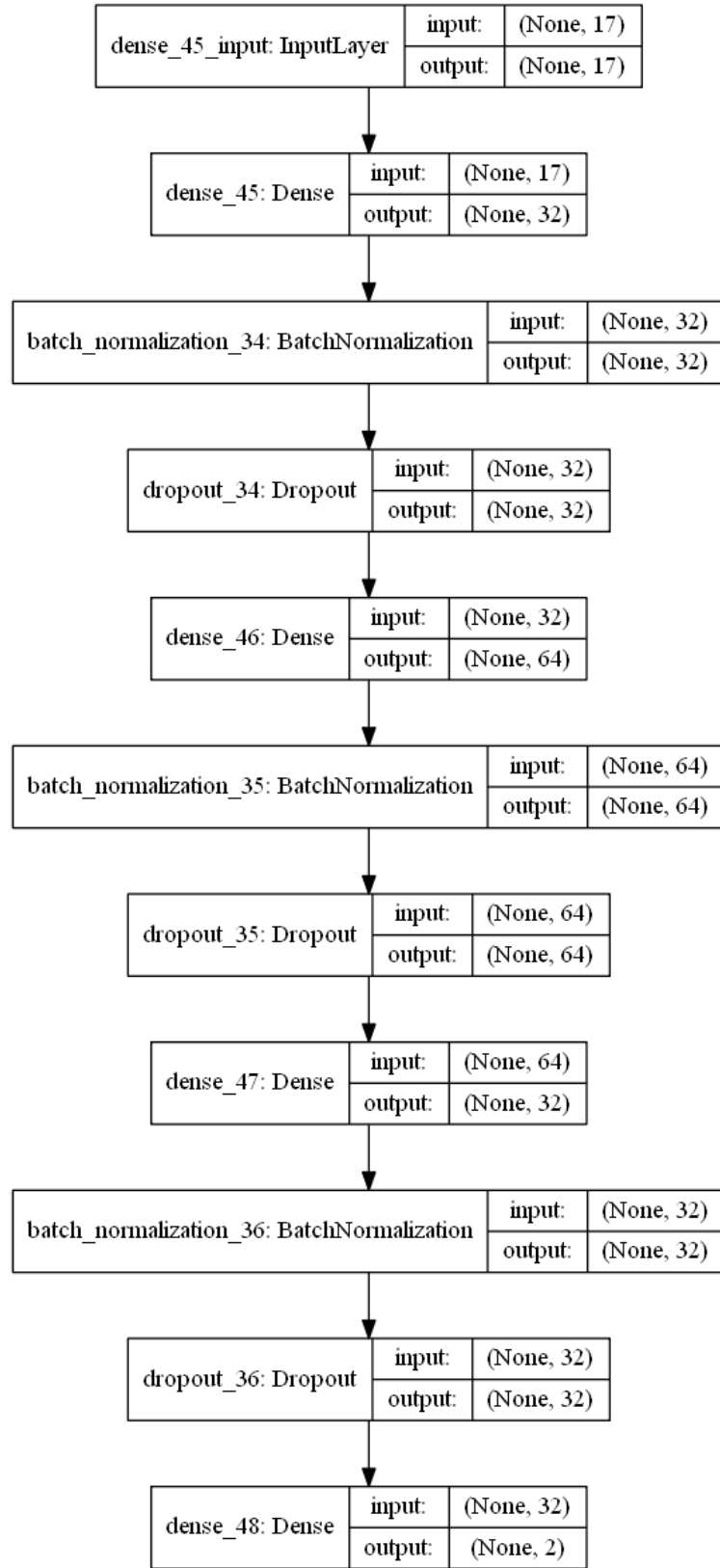


Figure 17: Multi Layer Perceptron Model scheme (trained on 17 variables)

## G Training the MLP : additional plots

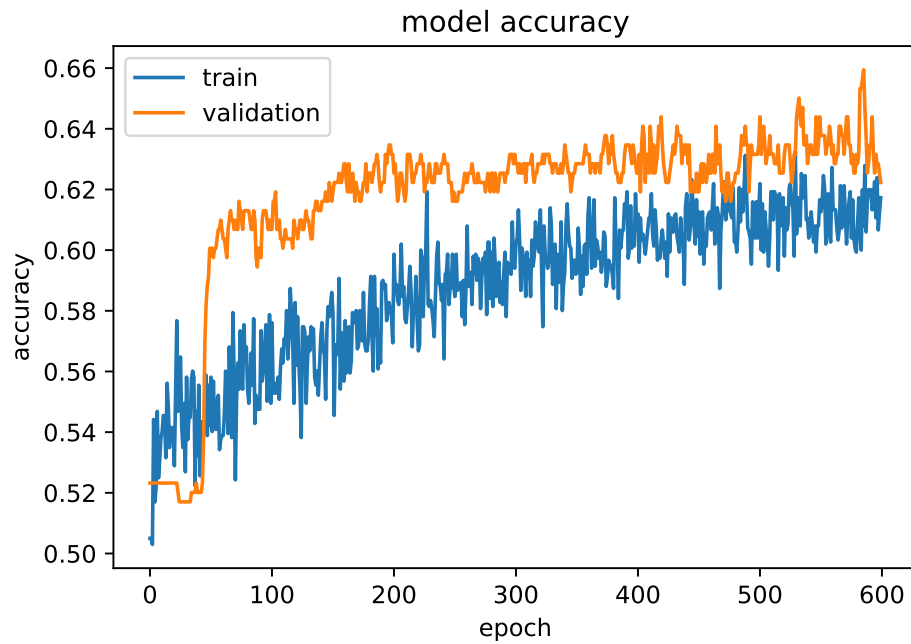


Figure 18: Evolution of the accuracy during the training of the 15 variables MLP

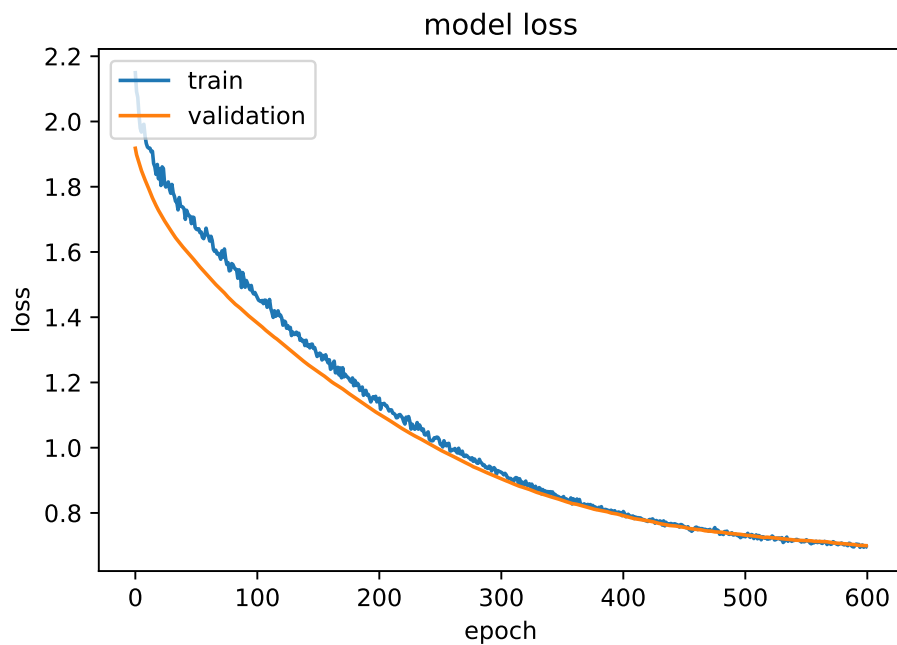


Figure 19: Evolution of the sparse categorical crossentropy during the training of the 15 variables MLP